

# Administration de machines Unix dans plusieurs fuseaux horaires

Stéphane Bortzmeyer

<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 16 décembre 2011. Dernière mise à jour le 17 décembre 2011

<https://www.bortzmeyer.org/date-utc.html>

---

Si vous administrez des machines Unix situées dans plusieurs fuseaux horaires, vous vous êtes peut-être déjà posé la question : quel fuseau indiquer à la machine ? Celui de sa localisation physique ? Celui de **votre** localisation physique ? Un autre ?

Voici la situation : Jean Michu, administrateur système est à Paris et il gère des machines à Newark, Saint-Louis et d'autres endroits. Sur Unix, on peut configurer chaque machine pour indiquer son fuseau horaire (je ne crois pas qu'il existe de moyen standard, par contre, sur Debian, c'est `dpkg-reconfigure tzdata` et sur Red Hat, c'est `vi /etc/sysconfig/clock`). Mais lequel indiquer ? Il y a au moins trois solutions :

- La localisation physique de la machine. Cela peut être plus pratique lorsqu'on veut communiquer avec des humains qui sont sur place, par exemple pour coordonner une intervention. Mais, autrement, cette localisation est une des choses les moins importantes dans l'Internet d'aujourd'hui. Avec le nuage, on peut même ne pas savoir dans quel fuseau horaire se trouve la machine qu'on administre. En outre, cela rend plus difficile de comparer des estampilles temporelles sur les différentes machines (par exemple pour déterminer si elles ont eu le même problème au même moment).
- La localisation physique de l'administrateur système. Cela a certainement plus de sens que celle de la machine : contrairement aux ordinateurs, les humains sont fortement dépendants de l'heure (la machine se moque de travailler à deux heures du matin ou à quatre heures de l'après-midi). Mais l'administrateur peut se déplacer. S'il se trouve temporairement en Chine, il va devoir jongler entre le fuseau horaire de sa localisation habituelle et celui de sa localisation actuelle. Et, surtout, s'il y a **plusieurs** administrateurs système, qui peuvent être dans des fuseaux horaires différents, cette méthode ne marche plus du tout.
- UTC ? Ce fuseau horaire a l'avantage d'être normalisé, et d'avoir un sens pour tout le monde quel que soit sa localisation sur la planète. Lorsqu'il faut communiquer avec d'autres personnes (par exemple parce qu'il y a eu un problème de sécurité et qu'il faut indiquer le contenu d'un journal), UTC est la seule référence internationale.

C'est donc la dernière solution que j'ai choisie. Je configure toutes mes machines de manière à ce que leur heure par défaut soit UTC, ce qui produit des journaux utilisant cette heure.

Mais n'est-ce pas pénible que la commande `date` donne cette heure UTC qui ne correspond pas au vécu de l'humain? Et que `ls-l` ne donne pas l'heure légale? Heureusement, Unix a réglé le problème depuis longtemps. Il suffit à chaque administrateur système de définir la variable d'environnement `TZ` et il aura l'heure dans **son** fuseau horaire :

```
% date
Fri Dec 16 20:03:03 UTC 2011

% export TZ=Europe/Paris
%

% date
Fri Dec 16 21:03:08 CET 2011
```

Même chose pour `ls`. Si les États-Uniens font parfois preuve de provincialisme (par exemple en utilisant ASCII sans penser que sept bits ne suffisaient pas pour toutes les écritures du monde), le fait que leur pays compte plusieurs fuseaux horaires a certainement contribué à mettre en place une gestion correcte de ce concept sur Unix.

Évidemment, aucune solution n'est parfaite. Par exemple, si on veut configurer `cron` pour lancer une tâche à une heure légale particulière, il faudra faire un peu de calcul avant de le programmer. Ceci dit, vous pouvez demander à `date` de le faire pour vous (merci à Gabriel Kerneis pour le rappel). Si vous voulez savoir quelle est l'heure légale à Douala lorsque UTC est à midi :

```
% TZ=Africa/Douala date --date="2011-12-16 12:00:00Z"
Fri Dec 16 13:00:00 WAT 2011
```

(Le format utilisé est celui du RFC 3339<sup>1</sup>, Z signifiant UTC.)

Si on veut faire l'inverse, trouver quelle sera l'heure UTC correspondant à une certaine heure légale (ici, on se demande quelle sera l'heure UTC lorsqu'il est cinq heures du matin en Californie) :

```
% TZ=UTC date --date="$(TZ=America/Los_Angeles date --date='2011-12-16 05:00:00')"
Fri Dec 16 13:00:00 UTC 2011
```

Une dernière chose sur les fuseaux horaires. Certaines personnes disent que l'argument de communication (« *We observed an abnormal traffic from your AS around 0200 UTC* ») n'est pas si important que ça car on peut toujours indiquer le fuseau horaire explicitement, même lorsqu'il n'est pas UTC (« *We observed an abnormal traffic from your AS around 0300 CET* »). Le problème est que ces abréviations ne sont pas forcément connues mondialement (demandez à un États-unien ce qu'est CET et à un Français à quoi correspond MST) et qu'elles sont ambiguës : par exemple EST peut être un fuseau horaire aux États-Unis ou bien en Australie. On peut demander à `date` d'afficher l'heure avec un fuseau horaire explicite, indiqué numériquement, sans ses abréviations (ici, pour le fuseau horaire de la Californie) :

1. Pour voir le RFC de numéro NNN, <https://www.ietf.org/rfc/rfcNNN.txt>, par exemple <https://www.ietf.org/rfc/rfc3339.txt>

```
# Par défaut, affiche une abréviation ambiguë
% date
Sat Dec 17 14:16:40 PST 2011

# Avec un décalage numérique, tout est plus clair
% date --rfc-3339=seconds
2011-12-17 14:16:46-08:00
```

mais les utilisateurs n'y pensent pas toujours.